A Note on Collapsing Bounded Query Classes

AUTHOR(S):

IZUMI, Masa-aki; KOBAYASHI, Kojiro

CITATION:

IZUMI, Masa-aki ...[et al]. A Note on Collapsing Bounded Query Classes. 数理解析研究所講究録 1988, 666: 61-70

ISSUE DATE:

1988-07

URL:

http://hdl.handle.net/2433/100685

数理解析研究所講究録
666巻 1988年 61-70

A Note on Collapsing Bounded Query Classes

by

Masa-aki IZUMI (和泉　正明)

Department of Management and Informatics

SANNO College (産業能率大学)

1573 Kamikasuya, Isehara, Kanagawa 259-11 JAPAN

and

Kojiro KOBAYASHI (小林　孝次郎)

Department of Information Sciences

Tokyo Institute of Technology (東京工業大学)

O-okayama, Meguro-ku, Tokyo 152 JAPAN

Abstract. We show a general theorem on collapsing bounded query classes. From this theorem, we derive some new results, for examole, $P^{NP[\log n]} = L^{NP_1[\log n]}$, etc.

1. Introduction

Recently several results about the collapse of certain oracle hierarchies have been proved (see [Hem 87], [Kad 87], [SW 87] and [Wag 87]). For example, in [Hem 87] it was proved that $P^{NE} = NP^{NE}$ and hence the strong exponential hierarchy $SEH = E \cup NE \cup NP^{NE} \cup NP^{NP^{NE}} \cup \ldots$ collapses ($E = \bigcup_{c>0} DTIME(2^{cn})$, $NE = \bigcup_{c>0} NTIME(2^{cn})$). These results have been proved by using census functions and the proofs are rather complicated.

In the present paper, we show one general theorem on collapsing complexity classes with restricted number of queries to the oracle. This theorem gives some new results concerning some well-known complexity classes. The general theorem is proved briefly by using the power of hard sets for certain complexity classes. We do not use census

2. Preliminaries

We mostly follow standard terminology and notations for complexity theory (see, e.g., [HU 79]). We assume that the alphabet Σ contains at least two letters 0, 1. Let Σ^* denote the set of all finite strings consisting of letters in Σ. A subset of Σ^* is called a *language*.

For x in Σ^*, let $|x|$ denote the length of x. The empty string is denoted by e, $|e| = 0$.

Our model for computation is oracle machines. A *nondeterministic*

(resp., *deterministic*) *oracle machine* M is a nondeterministic (resp., deterministic) multitape Turing machine with a *query tape* and three special internal states called QUERY, YES and NO. Let $M^A(x)$ denote an oracle machine M that is given x as an input and A as an oracle and let $M(x)$ denote $M^{\emptyset}(x)$. When $M^A(x)$ enters the state QUERY, the machine asks the oracle A whether the string written on its query tape belongs to A or not. If the string is in A, then the machine enters YES. Otherwise, it enters NO. A *transducer* is a Turing machine with a read only input tape and a write only output tape.

The notions of time bounded oracle machines and space bounded oracle machines are defined as for the usual Turing machines. For space bounded oracle machines, we do not bound the input tape and the query tape.

For a space bound S and an oracle A, let $\mathrm{NSPACE}(S)^A$ ($\mathrm{DSPACE}(S)^A$) denote the class of languages accepted by nondeterministic (resp., deterministic) oracle machines that have A as the oracle and that operate within space bound $S(n)$, and let NSPACE(S) (DSPACE(S)) denote $\mathrm{NSPACE}(S)^{\emptyset}$ (resp., $\mathrm{DSPACE}(S)^{\emptyset}$).

For a time bound T such that $T(n) \geqq n$, and an oracle A, let $\mathrm{NTIME}(T)^A$ ($\mathrm{DTIME}(T)^A$) denote the class of languages accepted by nondeterministic (resp., deterministic) oracle machines that have A as the oracle and that operate within time bound $T(n)$, and let NTIME(T) (DTIME(T)) denote $\mathrm{NTIME}(T)^{\emptyset}$ (resp., $\mathrm{DTIME}(T)^{\emptyset}$).

We define the classes of languages discussed in this paper.

Let L^A, ..., $PSPACE^A$ be the classes

$$L^A = DSPACE(\log n)^A,$$

$$DCSL^A = DSPACE(n)^A,$$

$$P^A = \bigcup_{i \geq 1} DTIME(n^i)^A,$$

$$NP^A = \bigcup_{i \geq 1} NTIME(n^i)^A,$$

$$NP_1^A = \bigcup_{c>0} NTIME(cn)^A,$$

$$E^A = \bigcup_{c>0} DTIME(2^{cn})^A,$$

$$NE^A = \bigcup_{c>0} NTIME(2^{cn})^A,$$

$$EXP^A = \bigcup_{i \geq 1} DTIME(2^{n^i})^A,$$

$$NEXP^A = \bigcup_{i \geq 1} NTIME(2^{n^i})^A,$$

$$PSPACE^A = \bigcup_{i \geq 1} DSPACE(n^i)^A$$

$$= \bigcup_{i \geq 1} NSPACE(n^i)^A \text{ (by Savitch's Theorem [Sav 70]).}$$

Let L, ..., PSPACE denote $L^{\emptyset}$, ..., $PSPACE^{\emptyset}$ respectively.

If A and B are subsets of Σ^* and $i(n)$ is a function, we say that A is *$i(n)$-space many-one reducible* to B ($A \leq_{i(n)\text{-space}} B$) if there is a deterministic transducer M that computes a function $f(x)$ within space $i(|x|)$ such that, for any x in Σ^*, $x \in A$ if and only if $f(x) \in B$. For any class $\mathcal{C}$, a set C is a *$\leq_{i(n)\text{-space}}$-hard set for $\mathcal{C}$* if $A \leq_{i(n)\text{-space}} C$ for any set A in $\mathcal{C}$.

3. The main theorem

In this section, we state and prove our main theorem. First we define some notations.

Definition 3.1. Let $q(n)$, $s(n)$ and $t(n)$ be functions. For any oracle A, $\mathrm{DTIME}(t(n))^{A[q(n)]}$ ($\mathrm{DSPACE}(s(n))^{A[q(n)]}$) denotes the class of languages accepted by deterministic $t(n)$-time (resp., $s(n)$-space) bounded oracle machines that make at most $O(q(n))$ queries to A. For any complexity class $\mathcal{C}$, let $\mathrm{DTIME}(t(n))^{\mathcal{C}[q(n)]} = \bigcup_{A \in \mathcal{C}} \mathrm{DTIME}(t(n))^{A[q(n)]}$ and $\mathrm{DSPACE}(s(n))^{\mathcal{C}[q(n)]} = \bigcup_{A \in \mathcal{C}} \mathrm{DSPACE}(s(n))^{A[q(n)]}$.

In this definition, we assume $0 \leqq q(n) \leqq t(n)$ for DTIME and $0 \leqq q(n) \leqq 2^{c \cdot s(n)}$ for some c for DSPACE.

Now, we present the main theorem.

Theorem 3.2. *Let $f(n)$ ($\geqq \log n$), $g(n)$, $h(n)$ and $i(n)$ be functions and let $\mathcal{C}$ denote the class of languages accepted by Turing machines that firstly make an $h(n)$ time bounded deterministic computation and then a $g(h(n))$ time bounded nondeterministic computation. If A is $\leqq_{i(n)\text{-space}}$-hard for $\mathcal{C}$, then*

$$\mathrm{DTIME}(h(n))^{\mathrm{NTIME}(g(n))[f(n)]} \subseteq \mathrm{DSPACE}(\max\{i(n + O(f(n))), f(n)\})^{A[f(n)]}.$$

Proof. Let L be a language accepted by a $\mathrm{DTIME}(h(n))$ machine M using

an oracle $B \in \mathrm{NTIME}(g(n))$ and making at most $O(f(n))$ queries to B.

Consider the following procedure :

begin

{x is the input of length n to M}

$u := e$;

while $\langle x, u\rangle \notin C_1$ **do**

if $\langle x, u\rangle \in C_2$

then $u := u1$ {the query answer is 'yes'}

else $u := u0$ {the query answer is 'no'} ;

if $\langle x, u\rangle \in C_3$

then halt and accept

else halt and reject

end

Here C_1 = {$\langle x, u\rangle$: $M(x)$ does not query to B any more}, C_2 = {$\langle x, u\rangle$: $M(x)$ queries to B at least one more time and its answer is 'yes'} and C_3 = {$\langle x, u\rangle$: $M(x)$ does not query to B any more and accepts}. Here, for the j-th query to B ($1 \leqq j \leqq |u|$), $M(x)$ does not query to B and uses the j-th bit of u as the answer. Clearly, the languages C_1 and C_3 are in $\mathrm{DTIME}(h(n))$ and the language C_2 is in $\mathcal{C}$. Hence the procedure can use the set A as oracle (A is a $\leqq_{i(n)\text{-space}}$-hard set for $\mathcal{C}$).

Since M makes at most $O(f(n))$ queries to B, the length of u is

at most $c_1 \cdot f(n)$ for some constant c_1, and u can be represented within space $c_1 \cdot f(n)$. The length of $\langle x, u \rangle$ is at most $n + c_1 \cdot f(n) + c_2$ for some constant c_2. Hence the reductions from C_1, C_2, C_3 to A can be performed within space bound $i(n + c_1 \cdot f(n) + c_2)$. Note that it is not necessary to represent $\langle x, u \rangle$ on a working tape as the input to the reduction because x and u are on the input tape and a working tape. Hence, our procedure operates within space $\max\{i(n + c_1 \cdot f(n) + c_2), c_1 \cdot f(n)\}$ and consequently within space $\max\{i(n + c_3 \cdot f(n)), f(n)\}$ for some constant c_3. It is obvious that the procedure makes at most $O(f(n))$ queries to A. Thus, $L \in \mathrm{DSPACE}(\max\{i(n + c_3 \cdot f(n)), f(n)\})^{A[f(n)]}$. Therefore,

$$\mathrm{DTIME}(h(n))^{\mathrm{NTIME}(g(n))[f(n)]} \subseteq \mathrm{DSPACE}(\max\{i(n + c_3 \cdot f(n)), f(n)\})^{A[f(n)]}. \quad \square$$

4. Consequences of the main theorem

Using universal sets and appropriate paddings, we can easily construct hard sets for many complexity classes.

Lemma 4.1.

(1) *There exists* $\leqq_{\log n\text{-space}}$*-hard sets for* NP *and* NEXP *in* NP_1 *and* NE *respectively.*

(2) *There exists* $\leqq_{n\text{-space}}$*-hard sets for* NE *and* $\bigcup_{c>0} \mathrm{NTIME}(2^{2^{cn}})$ *in* NP_1 *and* NE *respectively.*

(3) *There exists* $\leqq_{n^{i+1}\text{-space}}$ *-hard sets for* $\mathrm{NTIME}(2^{2^{n^i}})$ *in* NP_1 *and* NE *respectively.*

Proof. We can easily prove the lemma by using two sets

$L_0 = \{\langle i, x, 0^m\rangle$: Within time bound m, the universal Turing machine finds that the nondeterministic Turing machine i accepts $x\}$,

$L_1 = \{\langle i, x, 0^m\rangle$: Within time bound 2^m, the universal Turing machine finds that the nondeterministic Turing machine i accepts $x\}$

in NP_1 and NE respectively. □

Using Lemma 4.1 we have the following corollary of Theorem 3.2.

Corollary 4.2.

(1) $P^{NP[\log n]} = L^{NP_1[\log n]}$.

(2) $P^{NEXP[\log n]} = L^{NE[\log n]}$.

(3) $E^{NP[n]} = DCSL^{NP_1[n]}$.

(4) $E^{NEXP[n]} = DCSL^{NE[n]}$.

(5) $(\forall i \geqq 1)[\ EXP^{NP[n^i]} = PSPACE^{NP_1[n^i]}\ ]$.

(6) $(\forall i \geqq 1)[EXP^{NEXP[n^i]} = PSPACE^{NE[n^i]}\ ]$.

The class $P^{NP[\log n]}$ has been investigated recently in, for example, [Kre 86], [Kad 87], [SW 87] and [Wag 87]. Many of the languages related

to optimal solution size of NP optimization problems are members of $P^{NP[\log n]}$. For example, UOCLIQUE (Unique Optimal Clique), the set of undirected graphs that have one clique containing more vertices than each of other cliques of the graph, is one example of such languages (see, e.g., [Kad 87]). Thus, the class $P^{NP[\log n]}$ seems to be a very important natural class in the area between NP and P^{NP}. Now, Corollary 4.2 (1) shows that all of languages in $P^{NP[\log n]}$ are accepted within logarithmic space using $O(\log n)$ queries to a nondeterministic linear time oracle.

References

[Hem 87] Hemachandra, L., The strong exponential hierarchy collapses, Proc., 19 th Ann. ACM Symp. on Theory of Computing (1987) 110-122.

[HU 79] Hopcroft, J., and Ullman, J., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[Kad 87] Kadin, J., $P^{NP[\log n]}$ and sparse Turing-complete sets for NP, Proc. of Structure in Complexity Conf. (1987) 33-40.

[Kre 86] Krentel, M.W., The complexity of optimization problems, Proc. 18 th Ann. ACM Symp. on Theory of Computing (1986) 69-76.

[Sav 70] Savitch, W., Relationships between nondeterministic and deterministic tape complexities, *J. Comput. Sys. Sci.*, 4 (2) (1970) 177-192.

[SW 87] Schöning, U. and Wagner, K.W., Collapsing oracle hierarchies, census functions and logarithmically many queries, Tech. Report No. 140, Institute of Mathematics, University of Augsburg, April 1987.

[Wag 87] Wagner, K.W., Log Query Classes, Tech. Report No. 145, Institute of Mathematics, University of Augsburg, May 1987.